

SELECTING A TARGET DESTINATION USING SEEK COST
INDICATORS
BASED ON LONGITUDINAL POSITION

By

Brian Robert Pollock

Reed David Hanson

Nathan Boyd Wilson

Jason Dale Gregg

Mark Larry Birtzer

SELECTING A TARGET DESTINATION USING SEEK COST
INDICATORS
BASED ON LONGITUDINAL POSITION

5

Cross-Reference to Related Applications

This application claims the benefit of U.S. provisional application
Ser. No. 60/408,433 filed Sep. 5, 2002, titled "Enhanced Seek Distance
10 Determination."

Field of the Invention

This application relates generally to positioning systems, and more
15 particularly to such systems that decide the ordering of command
execution.

Background of the Invention

20 Computers and other types of host systems frequently transfer data
to and from nonvolatile storage devices such as disc drives. In modern
disc drives, storage capacity demands have increased at a dramatic rate,
which has caused great difficulties in the cost-effective manufacture of such
drives. A variety of schemes have arisen to cope with this challenge, some
25 of which complicate the operation of a disc drive. For example, some
modern disc drives are self-servowritten, by which it is meant that the
drive writes its own servo reference marks onto its data surface(s) with
little or no interaction with high precision servowriting machines. Other
modern disc drives use media that is pre-written, by which it is meant that
30 the servo marks are written before the discs are installed into the disc drive

-2-

housing. Both of these approaches introduce substantial misalignments that can affect the speed at which a given disc access command is performed.

5 In an ordinary operating mode the process of transferring data to or from a specific media location is initiated by the host system or device, which issues a command across a bus using a defined protocol. Some interface protocols, such as the Small Computer System Interface (SCSI) protocol, allow multiple commands to be queued in a memory within the disc drive. In other words, a read or write command can be issued from
10 the host and placed in queue without having to wait for the execution of any outstanding commands to be completed. In other systems, such queuing is carried out by the host.

Typically, disc drives are adapted to execute the commands in an order which is deemed most appropriate based upon the types of
15 commands in the queue. Intelligent disc controller firmware sorts the commands, using any of a variety of sorting algorithms, in order to reduce mechanical delays within the drive. Mechanical delays include the times required to seek, to spin to the beginning of a block of data, and to transfer the data associated with the command. Proper sorting of queued
20 commands can dramatically improve the performance of a disc drive data storage system by reducing these mechanical delays. For further background, see U.S. Patent 6,170,042, "Disc Drive Data Storage System and Method for Dynamically Scheduling Queued Commands," issued 2 January 2001 to Gaertner et al.

25 Unfortunately, no queue management systems exist that can account for misalignments such as those that exist in modern disc drives. What is needed is an effective system for taking such misalignments into account when deciding an order in which to access potential target destinations.

30

Summary of the Invention

A host or local processor is configured to select a target destination using lateral seek cost indicators that are each partly based on a
5 corresponding lateral offset indicator derived from a longitudinal position measurement. In one method embodiment, this improved selection process is used to obtain a more efficient ordering of data storage access commands. The (scalar) position measurement is obtained during or just after a prior command execution. Several seek lengths are estimated, each
10 corresponding to a queued command, a first one of the estimated seek lengths being "one" of the lateral seek cost indicators. After determining that the "one" lateral seek cost indicator corresponds to one of the queued commands that refers to a "non-ideal" target, a more desirable one of the queued commands is selected. (As used herein a "non-ideal" target is one
15 that is not reliably reachable within a system-defined major period such as nominal disc stack revolution time.)

In a first alternative method embodiment, each of the lateral seek cost indicators is calculated as an arithmetic combination of several terms, at least one of the terms being a sinusoidal or other periodic function of the
20 longitudinal position measurement. The lateral offset indicators are each used to adjust a "raw" seek length so as to determine an "effective" seek length for each of several queued commands. These effective seek lengths can be used to assess nearby potential short seeks with improved accuracy, so that an "apparently non-ideal" target can be selected or so that an
25 "apparently ideal" target can be passed over. Either of these types of corrections will improve queue execution performance.

In a second alternative method embodiment, a disc stack is configured to support pre-written data storage discs into a disc drive so that the target destination is a storage location on one of the discs. A
30 calibrated offset model is then derived based on the configuration, defining

how the longitudinal position measurement affects the lateral offset indicators.

In a third alternative method embodiment, a "source location" is identified by specifying a cylinder identifier, a head identifier, and a sector identifier, the sector identifier being the longitudinal position measurement. Many queued commands are received, each also including a cylinder identifier, a head identifier, and a sector identifier. A difference is computed between the source cylinder identifier and each of the target cylinder identifiers so as to obtain a preliminary seek length corresponding to each of the (longitudinally proximal) queued commands. Some of the queued commands are identified so that each corresponds to a preliminary seek length smaller than a given magnitude. The seek length corresponding to each of the identified commands is adjusted, the adjustments each being partly based on the corresponding identified command's target head identifier, on the source head identifier, and on the source sector identifier. (These "adjustments" are the lateral offset indicators, and the "adjusted indicators" include the lateral seek lengths.) Several latency indicators are then derived so that each corresponds to one of the queued commands, each of the latency indicators based on the corresponding command's target sector identifier and seek length. Some or all of the latency indicators are based on the adjusted seek lengths. Then one of the queued commands is selected for execution based on the latency indicators using as the target "destination" the selected command's target cylinder, head and sector identifiers.

A device embodiment of the present invention includes a disc stack and a controller configured to execute software-implemented methods described in this document. The disc stack has at least two rigidly supported, pre-written data storage discs. The controller is configured to select a target destination on one of the discs using lateral seek cost

-5-

indicators that are each partly based on a corresponding lateral offset indicator derived from a circumferential position measurement.

Another device embodiment further includes a target head and a source head, the source head being the one configured to generate the longitudinal position measurement. The target head is able to access the target destination, and the source head is not able to access the target destination. The controller includes a random-access memory that is constructed and arranged with enough space to contain a queue of more than 32 disc access commands that each include a target cylinder identifier, a target head identifier, and a target sector identifier.

Additional features and benefits will become apparent upon reviewing the following figures and their accompanying detailed description.

Brief Description of the Drawings

Fig. 1 shows a flowchart of a method of the present invention.

Fig. 2 shows a disc drive constructed to benefit from the present invention.

Fig. 3 shows a close-up "side view" from between two of the discs of the disc drive of **Fig. 2**.

Fig. 4 shows a "bottom view" of selected items in **Fig. 3** to illustrate one of the problems that has created the need for the present invention.

Fig. 5 shows a computer system implementing the present invention.

Fig. 6 shows a flowchart of another method of the present invention.

Fig. 7 shows a specific function coded in the "C" programming language, implementing one of the steps of **Fig. 6**.

Fig. 8 shows a table with specific values for use with the coded function of **Fig. 7**, the values calibrated for an actual disc drive.

Fig. 9 shows a seek performance model for the electromechanical system that accesses the disc stack of **Fig. 5** for reading or writing.

Fig. 10 shows a queue that contains commands to be manipulated in illustrating the operation of the present invention, and related computed values.

Detailed Description

Although the examples below show more than enough detail to allow those skilled in the art to practice the present invention, subject matter regarded as the invention is broader than any single example below. The scope of the present invention is distinctly defined, however, in the claims at the end of this document.

Numerous aspects of basic engineering and of positioning technologies that are not a part of the present invention (or are well known in the art) are omitted for brevity, avoiding needless distractions from the essence of the present invention. For example, this document does not articulate detailed and diverse methods for executing a seek, settle, track follow, or data transfer portion of a read or write command. Neither does it include implementation decisions such as whether virtual sector numbers are used to compensate for circumferential offsets between data discs. Specific techniques for constructing controller interface modules are likewise omitted, typically being a matter of design choice to those of ordinary skill in that field of technology.

Definitions and clarifications of certain terms are provided in conjunction with the descriptions below, all consistent with common usage in the art but some described with greater specificity. A "longitudinal" direction is aligned with sensor's nominal direction of motion in a given (stationary or moving) frame of reference. For example, a transducer following a track is moving longitudinally, whereas a transducer moves

-7-

“laterally” when seeking. A “lateral” direction is one that forms an angle of more than 45 degrees with the longitudinal direction, and typically more than 70 degrees. A seek “length” is a lateral seek cost indicator having units of length or track count increments.

5 Turning now to **Fig. 1**, there is shown a method **110** embodying the present invention, comprising steps **120** through **170**. Lateral seek cost indicators are generated based on a corresponding lateral offset indicator that is derived from a longitudinal position measurement **130**. Several seek lengths are estimated, each corresponding to a respective queued
10 command, one or more of the estimated seek lengths each being a respective one of the lateral seek cost indicators **140**. When it is determined that the one lateral seek cost indicator corresponds to one of the “non-ideal” queued commands **150**, another of the queued commands is selected for execution **160**, one that is reliably reachable within a predetermined
15 interval. After removing the executed command from the queue, steps **130** through **160** are repeated until there are no more “ideal” targets **170**. Using method **110** results in several commands being executed in rapid succession, especially for a deep queue containing commands with tightly clustered targets.

20 **Fig. 2** shows a “top view” of a disc drive **210** constructed to benefit from the present invention. (Terms like “top view” are arbitrary here, in that data handling systems like drive **210** can generally operate in any orientation.) Drive **210** includes “top” cover **223** that cooperates with base **202** to form a sealed chamber. Components supported in the chamber
25 include a spindle motor **215** which rotates a stack comprising one or more data storage discs **189,289** at hundreds or thousands of revolutions per minute. Information is written to and read from data surfaces on the disc(s) **189,289** through the use of an actuator assembly **261**, which rotates during a seek operation about a bearing shaft assembly **230**. Actuator
30 assembly **261** includes one or more actuator arms **290** which extend above

and below each of the disc(s) 189,289, with one or more flexures 293 extending from each of the actuator arms. Mounted at the distal end of each of the flexures is a head 134,234 that can fly in close proximity adjacent the corresponding data surface of an associated disc 189,289.

5 Servo and user data travels through a selected one of the heads 134,234 and flex cable 280 to control circuitry on controller board 206. (Controller board 206 is configured to interface with a host like disc controller unit 506 of Fig. 5 does, to perform a method of the present invention according to Fig. 1 or Fig. 6.) Flex cable 280 maintains an
10 electrical connection by flexing as each head 134,234 seeks along its path between tracks on disc(s) 189,289.

 During a seek operation, the overall track position of heads 134,234 is controlled through the use of a voice coil motor (VCM), which typically includes a coil 222 fixedly attached to actuator assembly 261, as well as one
15 or more permanent magnets 220 which establish a magnetic field in which coil 222 is immersed. The controlled application of current to coil 222 causes magnetic interaction between permanent magnets 220 and coil 222 so that coil 222 moves. As coil 222 moves, actuator assembly 261 pivots about bearing shaft assembly 230 and heads 134,234 are caused to move
20 across the surfaces of the disc(s) 189,289 between the inner diameter and outer diameter of the disc(s) 189,289.

 Difficulties have arisen in the cost-effective manufacture of data handling systems like that of Fig. 2. Many of the difficulties relate to exceedingly high track pitch and precise timing requirements. For
25 example, servo-writing many thousands of finely-pitched tracks takes a lot longer than servo-writing at lower densities. For this reason some manufacturers are looking to systems for installing pre-written discs into a data handling system. Others are looking to systems for having the data handling system servo-write itself. Both of these techniques can introduce

significant offsets between detections of marked positions not previously encountered.

To illustrate this Fig. 3 shows a close-up "side view" 399 from between two of the discs 189,289 of disc drive 210 (not to scale). As Fig. 3 shows, rotary actuator arm 290 supports read/write transducers 195,295, respectively positioned to access outer tracks 184,181 of disc 189 and/or from outer tracks 284 of disc 289. The pitch between tracks 184,284 is actually exceedingly fine, orders of magnitude denser than those shown. Read/write transducers 195,295 are supported by arm 290 via flexures 193,293, which are respectively equipped with microactuators 194,294. In implementing the present invention, fine control of the position of read/write transducers 195,295 is optionally made with the microactuator(s) 194,294.

Note that offset 198 is depicted in a circumferential direction relative to discs 189,289, which rotate on a spindle at a controlled speed about a common axis as shown by respective movement indicators 186,286. Offset 198 shows that transducer 195 leads transducer 295 slightly. Transducer 195 also happens to be closer to the discs' axis of rotation than transducer 295, as indicated by radial offset 197. Transducer 195 generates an output 177 that is received into buffer 178 of processor 288, which is implemented in control circuitry of controller board 206 of Fig. 2. Transducer 295 similarly generates an output 277 that is received into buffer 278. (Transducers 195,295 transmit outputs 177,277 via a preamplifier circuit supported on the actuator, not shown.)

An important component of transducer outputs 177,277 is position information found in servo wedges 181,281 on respective surfaces. (In Fig. 3, note that servo wedges 181 will not be encountered by transducer 195 at the same time that servo wedges 281 are encountered by transducer 295.) Most or all of the position information that enables transducers 195,295 stay on their respective tracks is found in marks within the servo wedges.

-10-

It should be understood that servo “wedges” are so named because they ordinarily taper narrower near the inner tracks of each surface. Also, the wedges are typically not perfectly radial. They each curve in a generally circular arc so that a rotation of actuator 330 will not greatly alter the time at which a given servo wedge will be encountered by a corresponding transducer.

This can be seen more easily in Fig. 4, a “bottom view” 191 of selected items in Fig. 3, in the rotating frame of reference 499 of disc 289. Arcuate servo wedges 265,266 are shown to extend between an outer diameter 269 and an inner diameter 267. Transducer 295 follows a nominally circular track 284 near the outer diameter 269, while transducer 195 zigzags along somewhere near offset circular track 184,185 of disc 189. Track 284 has a center 275 that is offset from the discs’ axis of rotation 276 by an offset 274 in a direction (phase) 273 as shown. Tracks 184,185 similarly share a center 175 that is offset from the discs’ axis of rotations 276 by an offset 174 in a direction (phase) 173 as shown. Offsets 174,274 are shown atypically large for clarity. For pre-written discs installed into a data handling system, it is expected that each track-center offset will be at least one to three orders of magnitude greater than a nominal track pitch. Centering errors of a similar magnitude may arise in field operation, particularly in laptop computers that suffer lateral shocks. Optionally the present invention includes steps of (1) detecting that such a centering error exists in periodic field calibration, and (2) responding so as generally to attenuate seek length estimation errors by recalibrating several values in a table (see table 800 of Fig. 8).

Recalling that Fig. 4 is a view from the discs’ frame of reference, it will be understood that transducers 195 & 295 seek or track follow radially as they rotate about the axis of rotation 276. At a selected moment of interest, transducer 195 is in position 411 and transducer 195 has just detected position 105. It is being de-selected, after which it will encounter

-11-

positions 106 and 107. Transducer 295 is in position 412, lagging and further out as shown in Fig. 3. Transducer 295 has just encountered position 236 without detecting it, is being activated, and is about to encounter and detect positions 237 and 238. Circumferential positions 100 through 152 on disc 189 are all encountered by transducer 195 during its rotation. Circumferential positions 200 through 252 on disc 289 are all encountered by transducer 295 during its rotation. All of these positions 100-152 & 200-252 are wedges that include a physical wedge number that is at least zero and at most 52. In modern hard disc drives there are hundreds of such wedges on each data surface.

Fig. 5 shows a computer system 500 implementing the present invention. Host CPU 504 is configured to communicate with a disc controller unit 506 to transmit data to or from a disc stack 507. Associated with disc controller unit 506 are disc buffer cache 510 (where data can be temporarily stored to shorten access time) and a local microcontroller 512 that controls unit 506.

Host CPU 504 executes system software 518, and communicates with the disc controller unit 506 through CPU system bus 514, disc interface adapter 516 that provides decode and electrical buffering, and disc interface cable 517. Disc controller unit 506 may be implemented in any number of ways, provided there is general similarity with the industry-accepted AT Attachment ("ATA"), SCSI, or other common interface protocol in effect as of this filing. The host interface protocol defines the interface between a host processor (e.g., CPU 504) communicating over a host bus 514 (e.g., a serial ATA bus) and a disc controller unit (e.g., unit 506).

Some of these interface protocols define a set of task registers, a disc interface cable connector, and associated interface signals. The defined task register set includes a command register containing the command code being sent to the disc drive, a data register for transferring data blocks

-12-

between the controller buffer and the host processor, registers concerned with the relevant disc drive address, sector and read/write head, and registers containing status information including error posting.

Some of these interface protocols also require certain real-time CPU-controller interface functions to be performed by the controller unit. For example, an inter-sector handshake may be required at each sector boundary on a disc, whereby the status of a previous sector must be posted before reading or writing from the next sector. Further, controller unit task file registers may require updating to indicate the identity of the sector currently being transferred. Finally, all write data may be automatically receipted for. One skilled in the art will recognize that many of these optional features of system 500 create synergies with the present invention that can be advantageous.

The example of system 500 is primarily a block interface. Host CPU 504 requests a block of data to or from the controller unit 506 in terms of a physical cylinder-head-sector (CHS) address, but does not specify or control where or in what form that data shall be stored in buffer cache 510, or on the disc stack 507. Buffer cache 510 is optionally implemented as Dynamic Random Access Memory ("DRAM").

As shown in Fig. 5, controller unit 506 includes a host interface unit 526 preferably comprising a microprocessor-controlled sequencer, or state machine 528 and a set of registers 530, including interface-compliant task registers. Unit 526 further includes buffer control unit 534 that communicates with the interface unit 526 via a First-In First-Out ("FIFO") buffer 532, that communicates with a disc sequencer 538 via a second FIFO buffer 536. Control unit 534 also communicates with the disc buffer cache 510. Disc sequencer 538 transmits data between disc cache 510 (via buffer control 534) and disc stack 507. Lines 533,535 transmit address and control information to and from buffer control unit 534 as shown.

Controller unit 506 also includes a local microprocessor interface 542 that communicates with the sequencer unit 528, and with a disc sequencer/state machine 538. The disc sequencer 538 is also coupled between FIFO 536 and the disc stack 507, and preferably also to an Error
5 Correcting Code ("ECC") logic unit (not shown) that corrects data read from the disc stack 507.

Within controller 506, data is stored in FIFO 536 before either transmission to disc stack 507 or to buffer cache 510. FIFO buffers 532 and 536 are each sized to hold less than one sector's worth of data. This
10 advantageously permits data stored in FIFO 536 to be read out while data are being written into FIFO 532. Buffer control 534 addresses and provides buffer cache read/write timing signals and controls to the buffer cache 510.

Upon command from the host processor 504, data to or from memory associated with the host CPU is parallel-transferred into the
15 controller unit 506 through the interface unit 526, FIFO 532, and then into the disc buffer cache 510. Upon receipt of a host CPU command, the controller unit local microprocessor 512 is interrupted, or the local microprocessor may simply be in a polling loop, continuously testing to see whether a host CPU command has been received. In either case, once an
20 incoming command is recognized, local microprocessor 512 then reads task file registers (among registers 530) and decodes the command and command bits from the command register.

Task file registers (among registers 530) within interface unit 526 are updated repeatedly during disc drive data transfer operations. As such,
25 the host system software 518 accesses the task file registers 530 to monitor data transfer and to associate errors reported by disc stack 507 with respect to particular sectors. Task file registers 530 specify the logical disc address to be read/written by the host system 504, 516, and will reflect the logical disc address of the sector currently being transferred to or from the host
30 system. Disc drive software 524 also accesses task file registers 530 as the

local microprocessor 512 supervises the disc stack 507, as it executes the host system commands, and as it directs transfer of data to and from disc cache 510 and the storage media within disc stack 507. Local microprocessor includes RAM 521 in which several items described below reside, as explained below with reference to **Figs. 6-10**.

Fig. 6 shows a method 600 embodying the present invention including steps 605 through 665. A disc stack rigidly supporting at least two pre-written data storage discs is constructed 610 and arranged into a disc drive so that "reference" (source) and (available potential) "target" heads can access data surfaces of the discs. A cylinder, head and sector numbers are determined for the reference head 615 and for the target heads obtained from several queued commands 625. Differences are computed between the reference cylinder number and each of the target cylinder numbers, generating a preliminary seek length for each of several queued commands 630. The commands having "apparently short" seeks are identified 635 using a threshold, typically at least 50 cylinders and at most 5000 cylinders. Seek lengths of the "apparently short" seeks are adjusted by adding a weighted sum of several terms 645. At least one of the terms is a sinusoidal or polynomial or other simple function of the reference sector number or similar longitudinal position measurement. The terms are weighted by some coefficients that depend on the reference and/or target head identifiers. A very detailed example scenario is explained below with reference to **Figs. 7-10**.

For each of several commands, a latency indicator is derived based on the command's target sector number and seek length 650. Some of these seek lengths are "adjusted" as described above. One of the queued commands is then selected for (immediate) execution based on the derived latency indicators, the selected command defining the target (destination) cylinder, head and sector 660.

Fig. 7 shows a specific function 700 coded in the "C" programming language, implementing step 645 of Fig. 6. The code includes program commands and other mathematical expressions (in remarks set off by "///
5 cosine functions are implemented in a table that is self-explanatory. The function "sa_EvalDeltaS" refers to a portion of RAM 521 called "Table.i16_DeltaS" containing 5 calibration constants (co through c4) for each "source" and "destination" head, to be explained next.

Fig. 8 shows a specific table 800 that exemplifies "Table.i16_DeltaS" as used in function 700 of Fig. 7. Column 880 shows the head number, each of the rows 810,811,812,813 containing the coefficients for the corresponding head. In terms of function 700, column 870 contains values for "c0," column 871 contains values for "c1," column 872 contains values for "c2," column 873 contains values for "c3," and column 874 contains
15 values for "c4." The use of these coefficients is most easily understood with reference to the example of Fig. 10, and particularly with regard to command 1013 shown there.

Fig. 9 shows a seek performance model for the electromechanical system that accesses disc stack 507 of Fig. 5 for reading or writing. Disc
20 stack 507 is accessed by an actuator assembly similar to assembly 261 of Fig. 2, but not shown in Fig. 5. Seek time 901 is shown in arbitrary, convenient units that are each equal to $1/N$ of a nominal disc stack revolution time, where N is the number of servo wedges on each data surface. Seek cost 902 is shown in arbitrary, convenient units that are each
25 equivalent to one nominal data track width. Profile 960 shows a read seek time within which a seek having a given cost will reliably be complete so as to permit data to be read from the disc stack. Profile 970 shows a write seek time within which a seek having a given cost will reliably be complete so as to permit data to be written to the disc stack. Model 900 predicts that
30 even a very short seek require about 20 servo wedges to settle reliably, and

about 28 servo wedges to settle reliably enough to permit writing. Both profiles 960,970 were derived empirically based on a given actuator assembly's performance. Both profiles 960,970 predict seek performance to at least about 99% certainty (i.e. so that less than 1% of the seeks at a given cost take longer than the time given by model 900).

Fig. 10 shows a queue 1000 that contains commands to be manipulated in illustrating the operation of the present invention. Each of the rows 1001 through 1038 represents one disc access command. Column 1051 indicates whether the command is a read (0) or a write (1). Columns 1052,1053,1054 respectively indicate the cylinder, head and sector of the command. For simplicity of the present example, the sector numbers designated in column 1054 are servo sector numbers that are assumed to align between data surfaces. In a typical modern implementation some sector number transformations may be appropriate.

Suppose that a prior command has just been completed and that the head that was used for it is a reference head, for present purposes. Disc stack rotation speed and seek speed are both effectively constant. The reference head's longitudinal position (sector) and lateral (track) position were known a very short (measured) time ago, during the just-finished execution. Therefore suppose it is known that the source location is cylinder 31000, head 2, and sector 44. The question is to determine which of the queued commands to execute next.

Note that it is not possible to execute command 1008 without a full disc stack rotation, which is a very substantial delay. This is because command 1008 requires a read from cylinder 30969, 31 tracks away from the source head's current position at cylinder 31000. Commands 1009,1010,1011 are also not feasible for a short latency (i.e. less than one disc rotation) because model 900 shows that a longitudinal delay of more than 20 sectors is always required for a reliable seek. Command 1012 has a target destination (sector) only 21 sectors away, indicating much too short

a time within which to execute a seek from cylinder 31000 to cylinder 14733. Command 1013 has a target destination 25 sectors away, which indicates a long enough duration to execute a very short read seek. By applying the method, function, table and model of Figs. 6-9, however, it
5 will become clear that command 1013 is not viable for a partial revolution seek.

Note that microprocessor 512 is configured to implement steps 615 through 660 of Fig. 6, and that all features of Figs. 7-9 are effectively stored in RAM 521 and available for use. At step 635 of method 600, several
10 commands are identified with "apparently short" seeks from "reference cylinder" number 31000. Those "identified" include commands 1013, 1014 and 1015 but not command 1012. (It may be convenient to consider each command in the sequence shown in Fig. 10, rather than to compute all "preliminary" seek lengths before deciding which command to execute
15 next.) Advantageously, method 600 provides a mechanism for executing the function of Fig. 7 selectively, only on "apparently short" seeks..

Values 1060 are computations that can optionally be performed on the values from queue 1000 in implementing the present invention. They can be performed in parallel and stored in a table, or they can be
20 performed successively so as to save memory. Column 1061 contains each apparent seek length. Column 1062 contains an adjustment to be added to the seek lengths of those seeks that are "apparently short." Column 1063 contains an adjusted list of several seek length values, only some of which are adjusted. Column 1064 contains an indication of whether a command's
25 target is reliable reachable within a predetermined longitudinal interval (of a partial or full revolution), for each of the commands containing a potential target within a quarter-revolution of the source head's most recently measured location.

-18-

Consider step 645 (as function 700) applied to command 1013 with reference head 2 at cylinder 31000 and sector 44. In the order of method 700, the "adjustment" to the seek length will be a sum of five terms:

- 5 The "c4" term will be $\text{COS}(2 * \pi * 44/288) * (-10 - 0) = -5.7$.
 The "c3" term will be $\text{SIN}(2 * \pi * 44/288) * (-32 - -2) = -24.6$.
 The "c2" term will be $(30973/16/2^{15})^2 * (-1775 - 248) = -7.1$.
 The "c1" term will be $(30973/16/2^{15}) * (43 - 402) = -21.2$.
 The "c0" term will be $(16 - -13) = 29$.

10

The adjusted seek length is this:

$$|30973 - 31000 + (-5.7 + -24.6 + -7.1 + -21.2 + 29)| = |-56.6| = 56.6.$$

- 15 Even for a read command, model 900 shows that a seek cost of 56.6 cannot reliably be accommodated within just 25 sectors. Command 1013 is accordingly not selected, despite its apparent seek length of only 27 cylinders. By similar calculations it can be shown that the adjusted seek length for command 1014 is this:

20

$$|31022 - 31000 + (-6.3 + -23.8 + -5.4 + -21.9 + 29)| = |-6.3| = 6.3.$$

- It is apparent from this calculation that although command 1014 is apparently an inward seek of only about 22 track widths, it is actually more
 25 like an outward seek of about 6 track widths. This is a very short and easy seek to accomplish in an interval 34 sectors long, according to model 900. By enhancing the accuracy of seek cost indicators, this example shows that the ordering of commands can be enhanced. The effect of this enhancement will be especially pronounced for queued commands that
 30 define a clusters of queued targets denser then 5 targets per 5 tracks, which

is denser than those of Fig. 10. It will also be more pronounced for a system with an actuator assembly with seek time models that require very short seeks to be complete and settled in less than one millisecond. This is faster than those of Fig. 9, for which 1 mS is about 17 units on the scale of seek time 901.

It is to be understood that even though numerous characteristics and advantages of various embodiments of the invention have been set forth in the foregoing description, together with details of the structure and function of various embodiments of the invention, this disclosure is illustrative only. Changes may be made in detail, especially in matters of structure and arrangement of parts within the principles of the present invention to the full extent indicated by the broad general meaning of the terms in which the appended claims are expressed. For example, the particular elements may vary depending on the particular position monitoring application while maintaining substantially the same functionality. Although the more detailed embodiments described above relate to data handling devices, other applications involving command sorting can readily benefit from these teachings without departing from the scope and spirit of the present invention. For example, although the above examples are typically oriented toward storing calibration data in a non-volatile memory during a manufacturing operation, such operations can also be used in a periodic or initial field calibration operation.

Moreover, it will be appreciated by those skilled in the art that the selection of a suitable combination of calibration memory size, accuracy, and formula complexity is a trade-off. The best solution will depend on the application, and except as specified below, no particular solution to this trade-off is of critical importance to the present invention. Moreover a selection of formulae will typically be available and readily derived, depending on the applicable geometry. One of ordinary skill will be able to use the above description to make and use a variety of polynomial- or

-20-

sinusoid-based or other implementations in light of the teachings above, without undue experimentation.